DETAILED ACTION

1. Claims 1, 10-13, 16, 17, 25, 27, and 29-36 are pending in this application with claim 1 has been amended, claims 2-9, 14, 15, 18-24, 26, and 28 cancelled, and claims 28-36 are newly added.

Examiner's Amendment

2. An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the issue fee.

Authorization for this examiner's amendment was given in a telephone interview with Joseph O'malley (Reg. No. 36,226) on 2/11/2009. The Examiner's amendment is necessitated to further clarify the claimed invention.

Claims 1 and 36 have been amended below.

Claims 18, 26, and 28 have been cancelled.

1. (Currently Amended): A computer program product including a storage medium with instructions thereon for execution by a computer for high level dynamic code generation, the instructions comprising:

a) computer code for dynamically generating at run-time a class file container object that stores source code describing a class file, wherein generating the class container object includes selecting a class name and a super class for the class, wherein dynamically generated code exists for the life of a server it resides upon;

b) computer code for adding a first source code defining a method to the class stored in the class file container object, wherein an application programming interface is used to define the method;

c) computer code for repeating step b for each method of the class;

d) computer code for adding a second source code into the method in the class stored in the class file container object, wherein an application programming interface is used to define code added into the method;

e) computer code for repeating step d to populate each method of the class stored in the class file container object;

f) computer code for generating a tree of statements and expressions based on the class stored in tile class file container object, wherein each statement and expression is represented as an object, wherein each statement maintains state of the program being generated, wherein the tree of statements and expressions forms a known structure when the class is at least one of an adapter and a proxy type, wherein organization of objects in a particular structure or interface avoids a need for a compiler;

g) computer code for using the tree of statements and expressions to generate byte code for the class; and

h) computer code for instantiating an instance of a new class file object from the byte code.

36. (Currently Amended): A system, comprising:

one or more processors; and

an application server including a dynamic code generation module with instructions for execution by the one or more processors, the instructions comprising:

a) computer code for dynamically generating at run-time a class file container object that stores source code describing a class, wherein generating the class file container object includes selecting a class name and a super class for the class;

b) computer code for adding a first source code defining a method to the class stored in the class file container object, wherein an application programming interface is used to define the method;

c) computer code for repeating step b for each method of the class;

d) computer code for adding a second source code into the method in the class stored in the class file container object, wherein an application programming interface is used to define code added into the method;

e) computer code for repeating step d to populate each method of the class stored in the class file container object;

f) computer code for generating a tree of statements and expressions based on the class stored in the class file container object, wherein each statement and expression is represented as an object, wherein each statement maintains state of the program being generated, wherein the tree of statements and expressions forms a known structure when the class is at least one of an adapter and a proxy type, wherein organization of objects in a particular structure or interface avoids a need for a compiler;

g) computer code for using the tree of statements and expressions to generate byte code for the class and

h) computer code for instantiating an instance of a new class file object from the byte code.

Examiner's Statement of Reason(s) for Allowance

3. The following is an examiner's statement of reasons for allowance:

The prior art of record i.e. Bentley et al. (US 5,815,415) relates to dynamically generating at runtime a class file for storing source code defining methods of the class. The source code is compiled into run-time code (i.e. bytecode) and object is instantiated from the generated class.

The prior art of record i.e. Bothner (US 6,110,226) relates to dynamically building a tree node representation of expressions, types, declarations, etc.

However, neither Bentley nor Bothner teaches *wherein each statement maintains state of the program being generated, wherein the tree of statements and expressions forms a known structure when the class is at least one of an adapter and a proxy type, wherein organization of objects in a particular structure or interface avoids a need for a* *compiler*. It would not have been obvious to one having an ordinary skill in the art at the time the invention was made to modify Bentley and/or Bothner to obtain the claimed invention.

Therefore, claims 1 and 36 are allowed over the prior arts of record.

Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

Correspondence Information

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Phillip H. Nguyen whose telephone number is (571) 270-1070. The examiner can normally be reached on Monday - Thursday 10:00 AM - 3:00 PM EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Wei Y. Zhen can be reached on (571) 272-3708. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

PN
2/11/2009

/Wei Y Zhen/
Supervisory Patent Examiner, Art Unit 2191